NASA / GODDARD SPACE FLIGHT CENTER

Advanced Architectures and Automation Branch

NGST Scientist's Expert Assistant (SEA)

Phase I Summary

October, 1997

Table of Contents

NGST Scientist's Expert Assistant (SEA)
Phase I Summary

Julie Breed, Jeremy Jones, Lisa Dallas, Sandy Grosvenor, Vijaya Krishnamurthy, Anuradha Koratkar, Jay Liebowitz

1 SUMMARY OF OBJECTIVES

1.1 PROJECT DESCRIPTION

The objective of the Scientist's Expert Assistant (SEA) project is to develop and evaluate visual and expert system (ES) tools to see if they can dramatically reduce the amount of manual effort that currently goes into the current "Phase II" General Observer proposal process for the Hubble Space Telescope (HST). To support this process, there are currently 30-35 staff members at the Space Telescope Science Institute (ST ScI) that support the final definition and review of Phase II proposals. For NGST, this staff must be reduced dramatically, primarily due to budget constraints. In addition to reducing manual effort, the SEA seeks to increase the quality of the resulting proposals. The project is a multi-phase, small team effort scheduled to take about 3 fiscal years. The primary phases of the project are summarized in the following paragraphs.

Phase I (end of FY97): The primary objectives of Phase I are to study, understand, and evaluate the existing HST process as compared with the expected NGST environment. This involves reviewing the areas that are most labor intensive and brainstorming visual and expert system approaches that might reduce that effort. Evaluating available ES development tools and environments for supporting development of working prototypes is also part of Phase I. Selecting an existing or upcoming actual space-based observing instrument on the HST that can serve as a "real-life" testbed for prototyping was also an objective of Phase I. The purpose of Phase I is to determine the scope of the project, and to develop a plan for prototype development.

Phase II (FY 98): The tasks for Phase II are to define and analyze requirements for the prototype. An initial design will be specified as well as the application framework. The requirements will be bounded.

Phase III (FY 99): The tasks for Phase III involve expanding the capabilities of the prototype to develop a fully functional tool for the test-bed environment. At completion, a working prototype should be completed and functional. We should be able to compare and evaluate the effectiveness of the prototype against the equivalent operational environment.

Phase IV (FY 00): Phase IV consists of implementing any additional enhancements, final wrap-up, review and analysis.

This report reviews and summarizes the results of Phase I.

1.2 PHASE I (LAST QUARTER OF FY 97) OVERVIEW

In Phase I, the primary focus has been to assess the project. We have focused on understanding the current proposal process, and have brainstormed ways to significantly improve the process by both reducing cost and increasing productivity. We have also been evaluating current development environments and tools. Since NGST is not targeted for launch for 10 years, we are mindful that our computing environment could see radical changes in the next ten years as it has in the last ten years. Consequently, while we are focusing on exploring new technologies, we are also trying to avoid locking into technology that may be obsolete by the time NGST launches.

Next, we have been refining the problem and solution set that we want the prototype to address. This includes setting the requirements and goals of the prototype tools and determining a specific set of tasks (or scenarios) that we want the tools to be able to handle. We want the prototype to be complex enough to realistically evaluate its potential, while simple enough that a small team will be able to successfully complete the project.

In addition to determining a set of tools and solutions to develop, we are also testing several user interface alternatives. Laying out these “straw man” interfaces has been one of the goals of Phase I.

Lastly, as for each phase of this project, we want to review progress to date, review our established set of success criteria, and re-focus the remaining Phases based on results obtained to date.

2 PROCESS AND TECHNIQUES USED

2.1 WEEKLY INTERVIEWS WITH HST/ ST ScI STAFF

The team has been meeting on a weekly basis with several staff scientists and programmers at ST ScI. We have been developing an understanding of both the Phase II proposal process for Hubble, and more specifically the areas of the process that consume the most amount of staff time. We have developed a high level "script" or description of the process and identified several areas in which we believe data visualization and expert system tools can make a substantial improvement.

2.2 IDENTIFICATION OF SPECIFIC SCRIPTS FOR ES RULE DEVELOPMENT

Early in this phase we selected the Advanced Camera for Surveys (ACS), a new instrument targeted for installation on HST in 1999 as the testbed for evaluating our prototype. The objective of selecting a live instrument is to have an environment that is "real" enough to give us an accurate sense of the complexity required to have a useful tool. While it would be desirable to have a NGST-based testbed, the development of NGST is not yet at a stage where we can accurately identify realistic instrument operational parameters. Further, with the timing of ACS's deployment, the prototype staff will have an opportunity to parallel the prototype software with the production proposal support software (namely, the Remote Proposal System 2 (RPS2)) to be developed by the STScI in support of ACS.

2.3 IDENTIFY PROTOTYPE SOFTWARE DEVELOPMENT ENVIRONMENT AND TOOLS

This step of the evaluation has involved reviewing the current technology and identifying the hardware and software environment, COTS packages, and development methodologies to be used for development of the SEA prototype. Taking advantage of the dramatic growth of the Internet and World Wide Web, we feel strongly that the tools should be system independent, and "delivered" via the Internet. Furthermore, we wish to take advantage of the maturing object-oriented methodologies to develop modules that are both independent and cooperative with each other.

For the expert system software, we have reviewed both government-developed and commercial packages available for developing, maintaining, and executing rules and have made a recommendation on the proposed package.

2.4 TRADE-OFFS AND CHALLENGES

2.4.1 Differences between HST and NGST

NGST is intended to be a far simpler observatory than Hubble. Simplifying the operations and reducing their cost is one of the primary design objectives of NGST. As such, many of the complexities that make detailing observing proposals for HST so complex will not be present in NGST. For example, the number of operating parameters and modes of the NGST's instruments are expected to be much simpler, and NGST's location in space will have far fewer constraints as does HST in its low earth orbit.

Consequently, while the team looks at ways to reduce the manual effort in HST observations, we are mindful that we want to avoid, where possible, spending effort on problems that will not be present in NGST.

However, in order to have our testbed environment realistic enough, it will be necessary to develop some features for the prototype tools that are unique to HST. We intend to employ sound object-oriented practices so that while there are, for example, some attributes of our ACS instrument module that are unique to ACS, the higher level "instrument object" will be not be unique to ACS.

2.4.2 Existing software vs. all new code

We are always mindful of the trade-off between using existing software, in whole or in part, and developing new software using new technologies. It is not the goal of this prototype effort to merely "port" existing software from Unix to the Web. While we believe that this would be useful, we do not think that it alone will accomplish the kinds of saving we are striving to achieve.

On the other hand, many of the subroutines that perform complex, detailed calculations pertaining to such aspects as planning and scheduling do already exist (for example, RPS2's 'synphot' routine) and it is sensible to use such software rather than spend effort re-writing it. There is and will continue to be that ongoing effort to balance new development versus linking to existing systems.

2.4.3 Synergy with other Observational Groups

There are several other projects underway to upgrade and improve user-interfaces for observatory proposal specifications. We know of at least two efforts currently underway: one with the Gemini project in Arizona and another with the AXAF group at Harvard. Part of our on-going research will be to stay in touch with these projects to see if there are ways we can work together to minimize overall software development effort and improve the overall value and flexibility of the prototype.

2.4.4 Value Added for all levels of Users

The SEA will be essential for novice users, and its assistance will be useful for the majority of the NGST user community. However, there will be some experienced users who may be bothered by 'assistance', if they already know exactly what they want. The SEA must accommodate these users as well. It will be a challenge to provide a rich environment which guides its users and hides confusing details from them, yet does not interfere with those users who wish to directly specify their parameters. The SEA team will work closely with the RPS2 developers, and will consider this their point of intersection.

3 ANALYSIS AND FINDINGS

Section 3 describes our findings as a result of our discussions with HST/STScI staff scientists. These include a description of the general Phase II specification process, the tasks associated with this process that are particularly troublesome or time consuming for observers and/or staff scientists, and the ways we recommend improving these tasks with regards to data visualization and/or expert system technologies.

3.1 GENERAL PROCESS SCRIPT

The current Phase II proposal specification process is essentially an iterative phase with five major steps.

3.1.1 Identify the target(s) to be imaged

Currently, a General Observer (GO) first manually researches their target for coordinates, attributes and other information, and then manually enters the information into their Phase II proposal. If necessary, the GO will manually enter an orientation for the target. When they do so, only the exact orientation itself is captured, the reasons they are specifying an orientation or a range of additional acceptable orientations are not captured. The Institute staff then independently validates target and orientation at least twice, with a final step including an image from the Guide Star Catalog sent to the general observer (GO) for confirmation just prior to execution of the proposal.

3.1.2 Identify the instrument and set the associated parameters

Currently, observers are responsible for understanding the instrument configurations in order to specify what instrument modes and camera filters to apply. After entering parameter values, the current system analyzes the parameters to look for conflicts and difficult-to-schedule options. Problems are referred back to the observer or the Institute staff to consider alternatives, and then amend and re-submit the proposal.

3.1.3 Determine the exposure for each image

The observer currently determines an exposures image through several means. They manually enter information into one or more of several STScI provided on-line exposure calculators which returns a set of tables or static graphs. They can look up information contained in hardcopy charts provided by the instrument manuals. And they can manually calculate their exposures using formulas contained in instrument reference manual. Regardless of the means, the observer must still manually enter their exposure times into their proposal and currently has no way of conveniently saving their analysis for later review and revision.

3.1.4 Review overall time requirements, adjust the above as necessary to ensure that the number of orbits required does not exceed those approved in the Phase I proposal

This step is an iterative process that may involve repeating the above steps. It involves ordering the exposures and "tweaking" the exposure times for a multi-exposure visit so that the total window of time is optimally used without going beyond the number of orbits that have been approved for the observations.

3.1.5 Specify the data output and delivery options

The last step in specifying the proposal includes deciding upon the data output and delivery options. This part has not been reported to be a source of significant manual effort.

3.1.6 Review and submit proposal

The final proposal is reviewed by the observer, then submitted to the Institute, and reviewed there by Institute staff. The observer and Institute staff work together to resolve remaining problems.

3.2 CURRENT HST PROCESS PROBLEM AREAS

Among the more manually intensive and time consuming areas of the current HST Phase II process, we have identified the following pieces that are targets for re-engineering and prototyping new approaches:

- Determining precise target coordinates. HST's instruments require a very high degree of accuracy in order to maximize the quality of the image. Currently, observers must separately research the various recorded locations of an object, and then manually enter the coordinates of the objects into HSTs proposal system software. It is not until the final checks before the observation is "flown" that graphical feedback is provided as to the expected field of view that will be imaged.

- Determining target orientation. This is an optional parameter that is often specified for a variety of reasons. Having a specific orientation is often desirable to control the impact of defraction spikes and bleeding or to manage the layout of spectroscopic slits. The problems that occur with specifying orientations are (a) typographical errors or errors in determining the desired orientation in the proper units, and (b) difficulty in scheduling the observation as the entire observatory must be rotated.

- Syntactical Errors. Since the current proposal format is eventually a text-based file with a fairly specific layout, basic syntactical and typographical errors are common and take time to track down and correct.

- Documentation is available but not integrated. Observers must separately consult several different large manuals in understanding the parameters of the instrument they wish to use and the process for developing a proposal for that instrument. While these manuals are available online, they are not readily cross-referenced and integrated with the proposal generation software.

3.3 GENERAL AREAS AND APPROACHES FOR IMPROVEMENT

Thus far, the team has focused on gaining an overview of the proposal process and defining the scope and schedule for the prototype effort. We have reviewed the overall Phase II process and the areas within it that are currently manually intensive. We have also begun developing the rules for prototyping an instrument configuration module for ACS. Finally, we are investigating similar development efforts for other observing platforms to see if we can leverage their work.

In the process of our analysis, we feel that there are five priority areas for development. These modules will initially be developed separately, but we expect that the tools will be integrated into a single iterative user-interface. The proposed modules are described below.

3.3.1 Graphical, "real-time" exposure calculator

This initial tool will generate real-time interactive graphs showing Signal-Noise Ratio and Source counts across a range of exposure times and wavelengths. The tool will allow the scientist to edit target or instrument parameters and instantly see their effect. We are targeting a Beta release for ACS by the end of December 1997. This tool is being developed in Java and will be a fairly simple application that should be able to replace the need to browse through over 100 pages of graphs and tables typically found in the HST Instrument Manuals. The development of this tool will also provide an initial development platform for the user-interface guidelines and underlying data objects that will be used for subsequent tools. We expect that in the first phase, this tool will not use expert system technology. Figures 1 through 8 show the layout of the prototype Calculator:

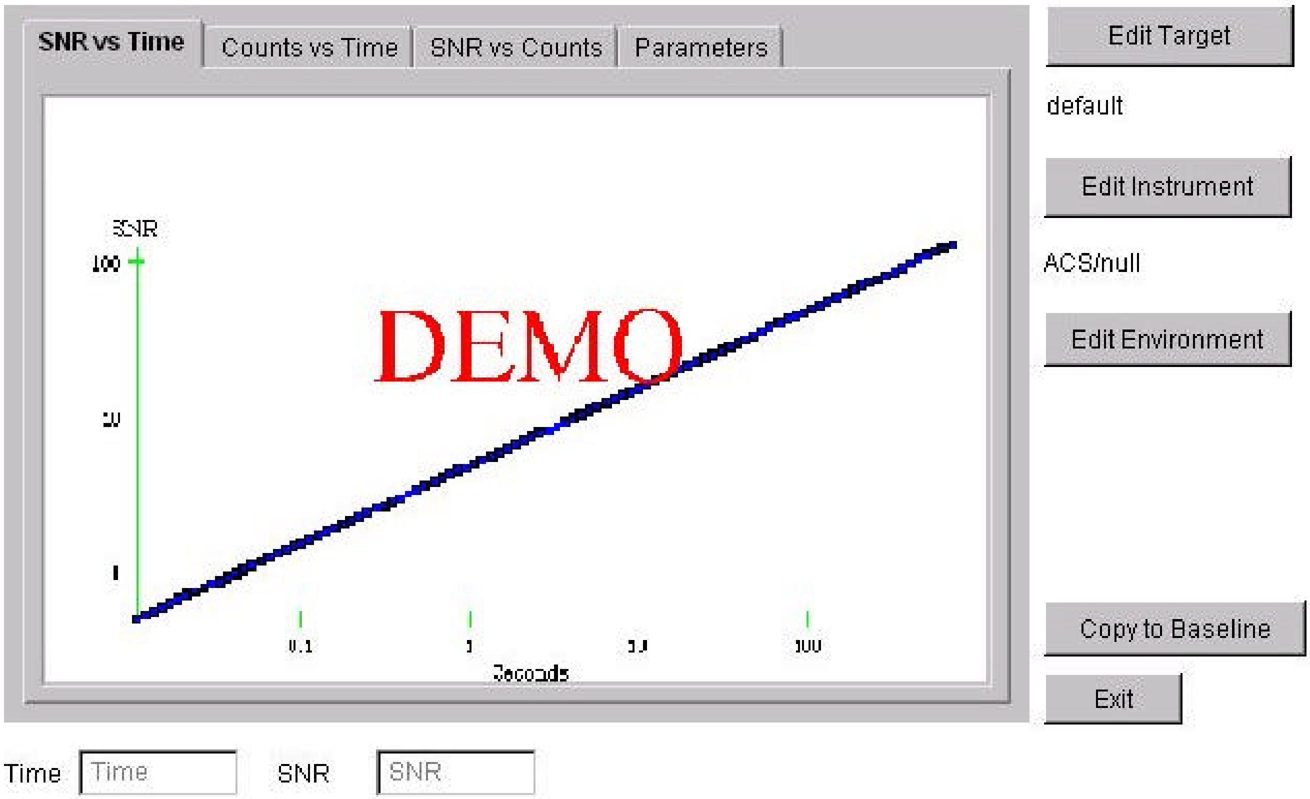

Figure 1: ETC Signal to Noise Ratio Over Time

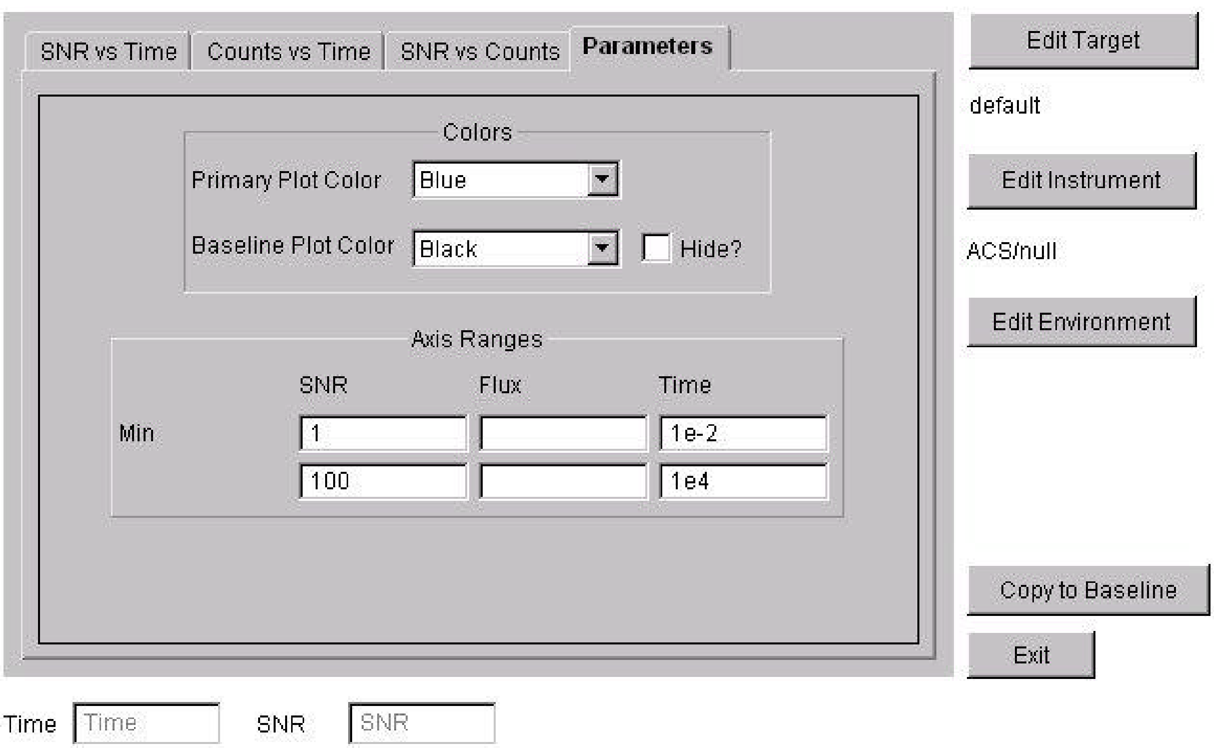

Figure 2: ETC Graph Parameter Specification

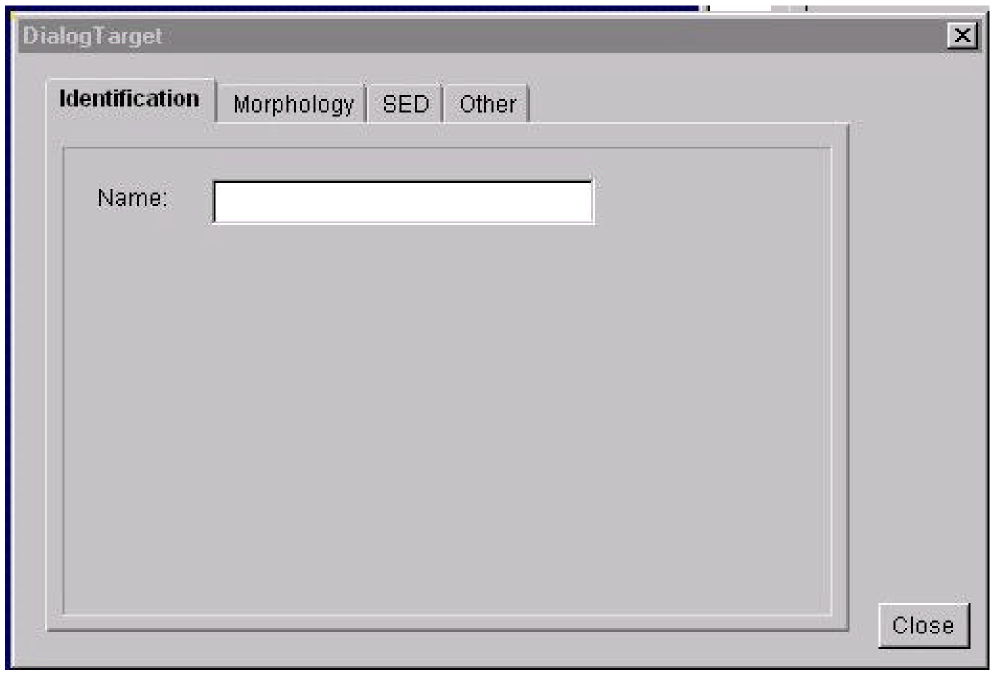

Figure 3: ETC Identification Tab Window

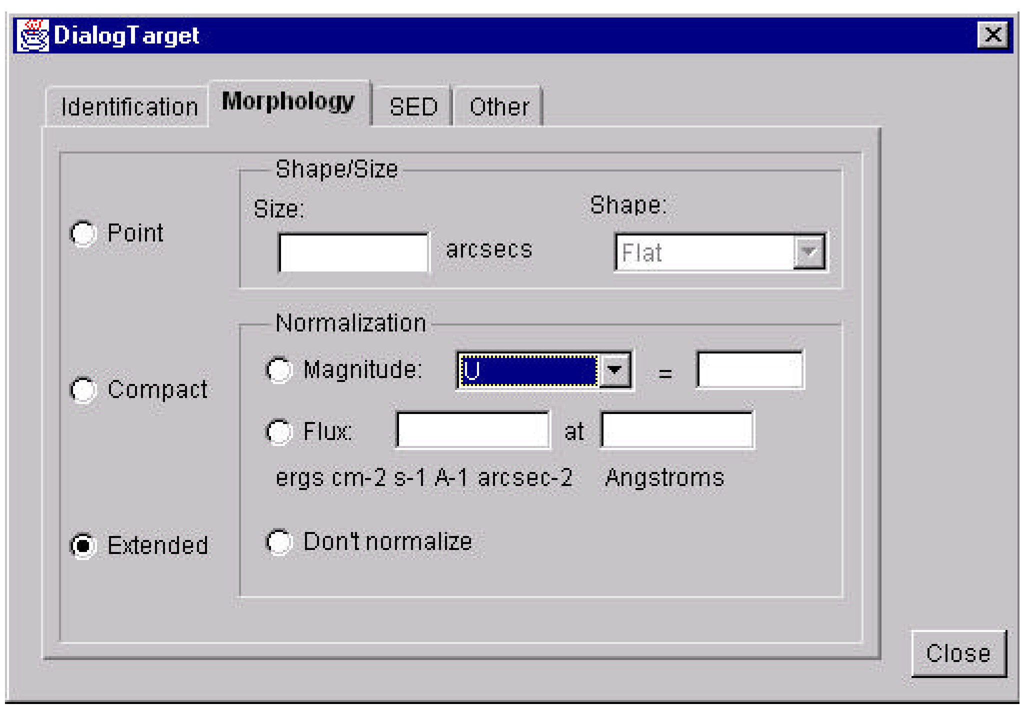

Figure 4: Morphology Tab Window

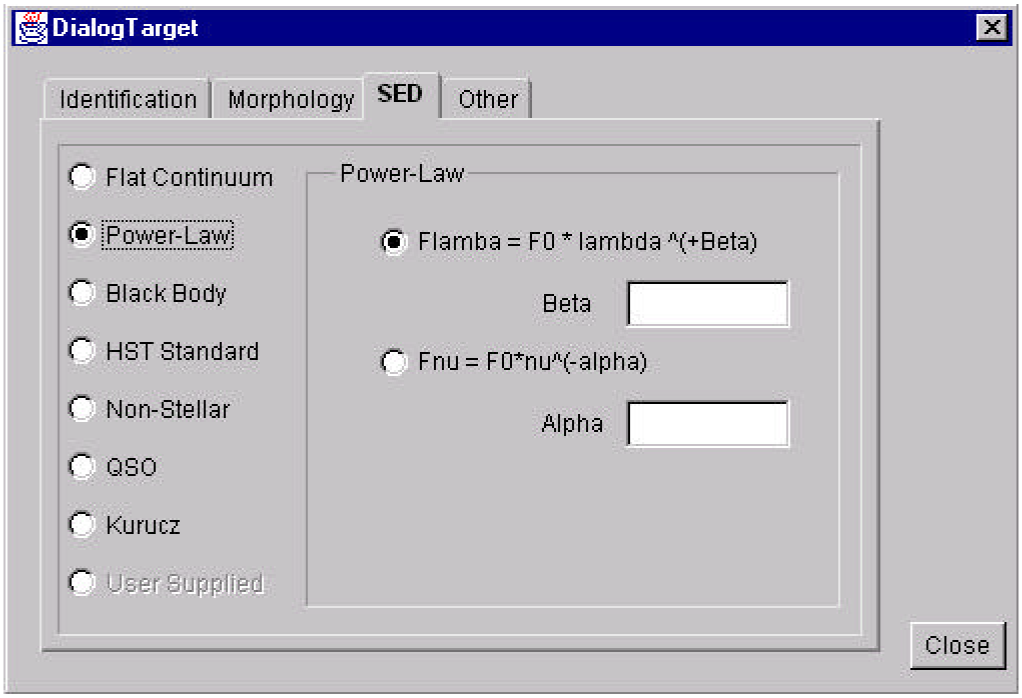

Figure 5: SED Tab Window

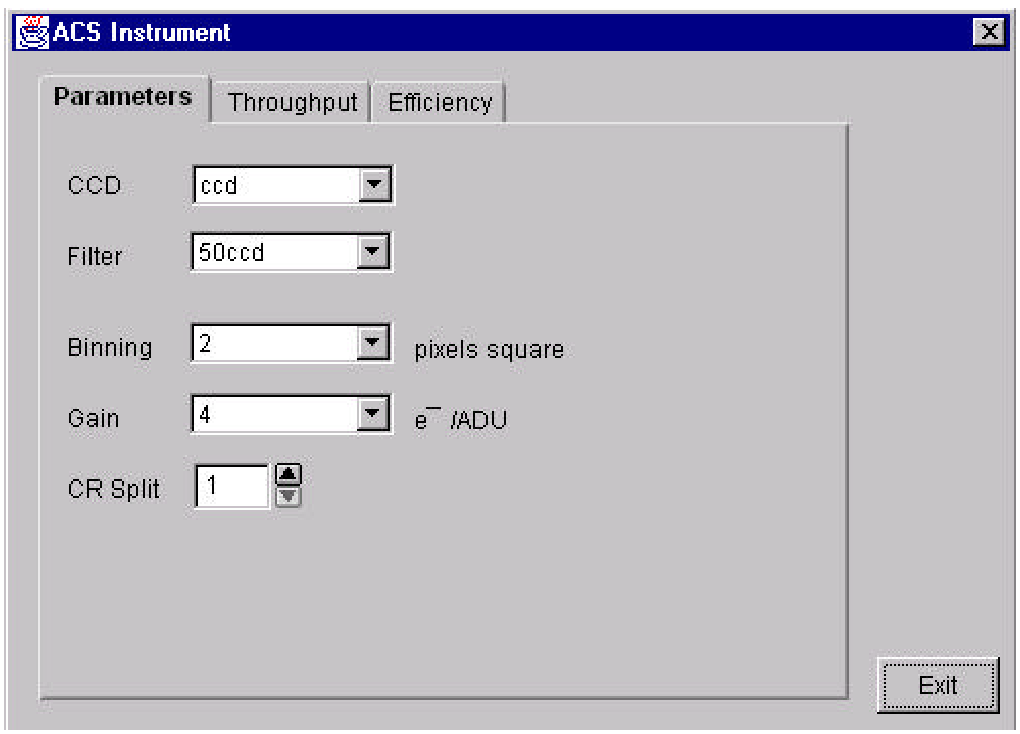

Figure 6: ETC Instrument Parameter Specification Window

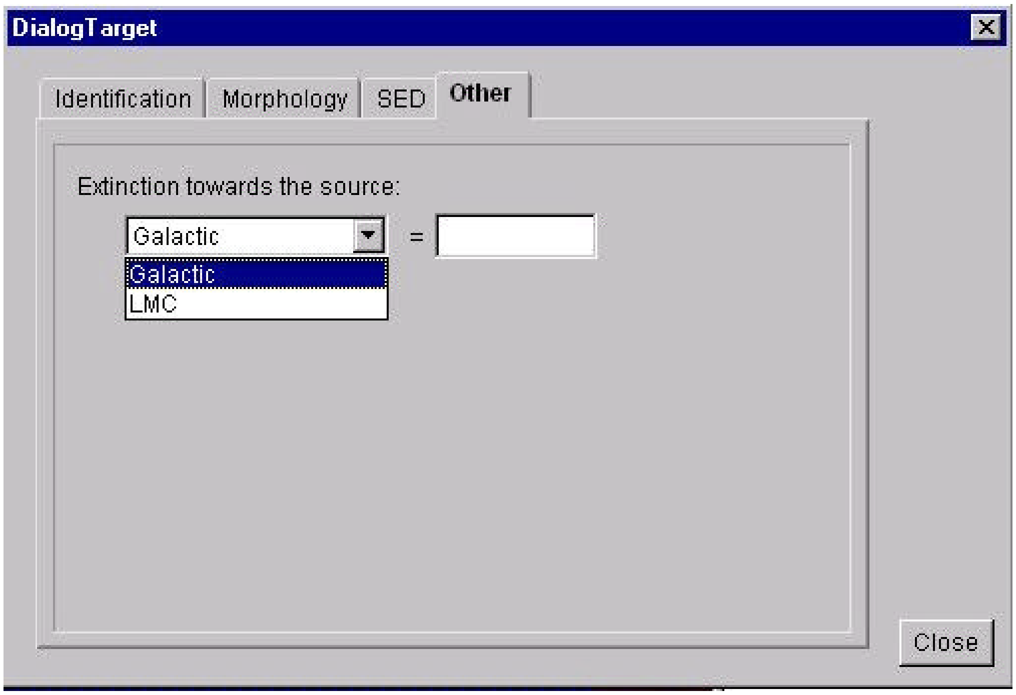

Figure 7: ETC Other Parameter Specification Window

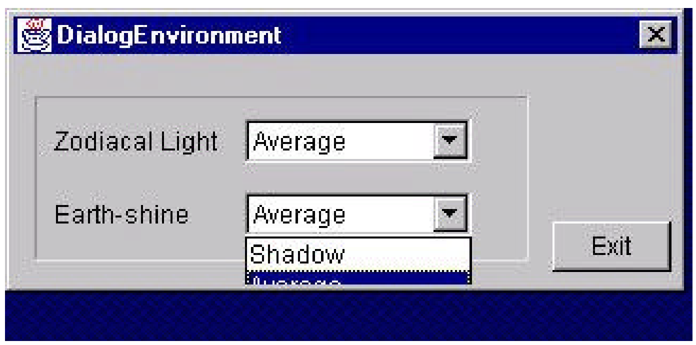

Figure 8: ETC Environment Window

3.3.2 "Visual" Target Tuner (VTT)

The VTT is a graphical tool for fine tuning target coordinates and orientation. Currently, observers must independently research target information and manually enter the information into their proposal. If they have a need to include or exclude specific objects, they must manually determine a specific orientation for the instrument. Giving a precise orientation requirement significantly hampers the schedulability of their program. The VTT will allow the user to easily specify the areas that need to be included or excluded and can therefore calculate a range of acceptable orientations to be passed on to the scheduling system. Further, there are currently no visual tools to help predict the overlap of spectroscopic slits, or the impact of refraction spikes.

The VTT seeks to be that visual environment. We are planning to prototype the VTT in several phases. The first phase (targeted for mid-1998) will be limited to displaying a retrieved FITS image, allowing the user to specify inclusion or exclusion areas, and fine tuning the specific location. In the second phase (targeted for late-1998) we will add the ability to model defraction spikes and spectroscopic slits. In both phases, we anticipate that this tool will be primarily a visual and graphical aid.

Figures 9 through 11 illustrate three proposed screens for the VTT.

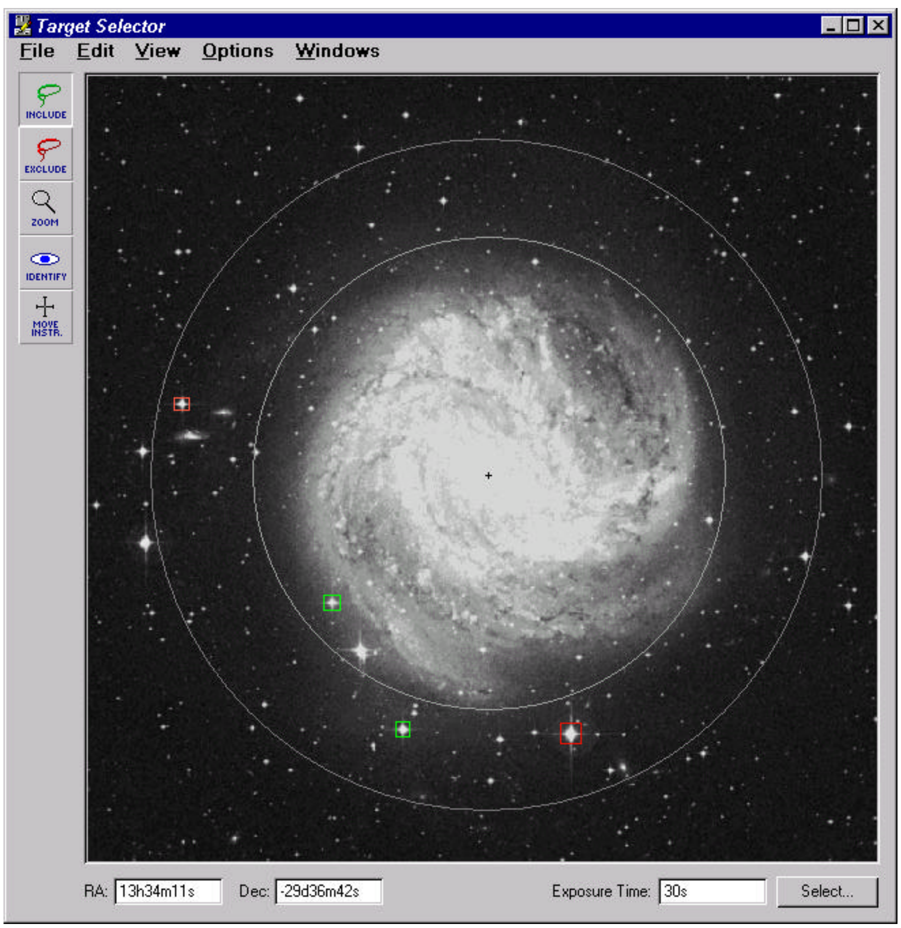

Figure 9: VTT Main Window

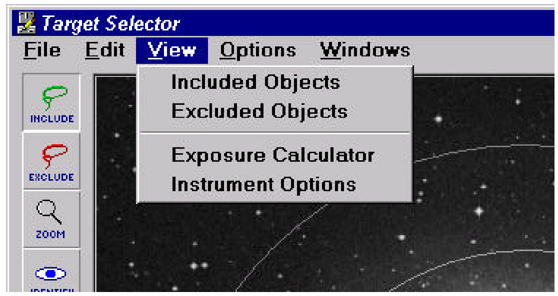

Figure 10: VTT View Menu

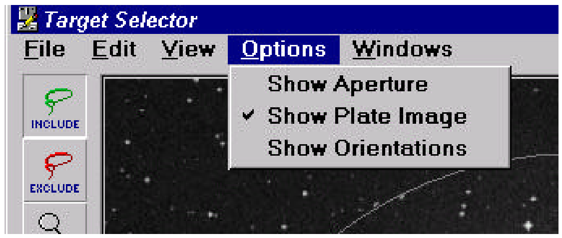

Figure 11: VTT Options Menu

3.3.3 Instrument Configuration Expert System (ICES)

This module will be a rule-based expert system that guides the observer through the definition of instrument parameters by asking a series of science-based questions, and provides recommended settings for the instrument based on the answers received. The goal is to eliminate the need for the observer to study and absorb the range of technical details about the workings of an instrument, and instead let them focus on the science they want to achieve. This tool will be developed in several phases. The first phase will have a fairly small rule-system that focuses on filter selection and emphasizes the development of a sound user-interface system and standards.

The user-interface for this expert system has some relatively unique needs. It needs to be able to integrate with the other tools and therefore have a compatible look and feel. It needs to be able to ask and save responses to questions in a manner that will be acceptable to both advanced and novice users. It needs to transparently interact with both the user on the "front-end" and a rules-engine on the "back-end". It also must support intelligent cross-references to technical literature, since while we are trying to allow observers to bypass up-front study of the technical instrument parameters, we are not trying to prevent them from studying the technical details. We want to help them focus quickly on the areas that are most relevant to their science objectives.

The second phase of the ICES will concentrate on expanding the rules and capabilities of the system. This is a critical objective for the SEA. We are striving to discover if the tool can contain a sufficient level of science expertise to free the observers from the technical details of the instrument and significantly reduce the support needed from Institute experts. We also must find out once such a system is achieved, if can we gain the acceptance from the observing community.

3.3.4 Visit Planner Expert System (VPES)

Thus far the modules described primarily focus on defining a single exposure, where the VTT is used to enter target parameters and the ICES is used to enter instrument parameters. The Visit Planner Expert System will work to provide assistance in laying out multi-exposure "visits". Both observing scientists and Institute staff currently spend a great deal of time planning multi-exposure visits. These challenges include:

- Laying out multiple exposures to create a mosaic
- Imaging a single target with a variety of instrument configurations.
- Planning not just the individual exposure times, but also the overhead time necessary to perform other tasks such as slewing the telescope, and reading the CCD buffers after an exposure.

These are currently manual, iterative processes that involve balancing exposure times to achieve the desired science objectives while keeping within the overall visit time constraints. The VPES is an expert system that will query the observer with a series of questions about their science objectives and priorities. It will recommend an optimal trade-off between individual exposure times and total visit execution time.

3.3.5 Re-validation Agent

This module has not yet been fully analyzed. Currently, the Program Coordinators (PCs) at the Institute spend a great deal of time re-processing already approved, but still pending proposals when a change to the instrument occurs. These changes can include a variety of things, for example, new calibration information that affects optimal exposure times. This concept for the re-validation agent is to use agent-based technology to evaluate the impact of changes to both submitted proposals and proposals that are still under development. The agent could seek out impacted proposals, calculate the effect of the impact, develop a recommendation and then notify the observer and Institute staff of possible alternatives.

3.3.6 User Interface Prototype

The following images include samples of the style of user interface that we are proposing for the NGST SEA. This is an extremely "rough" first cut at the high-level interface that will tie the modules together. The intent is that a General Observer's Phase II proposal will evolve from the original Phase I proposal, and that the components of the proposal will be grouped into a set of folders that can be manipulated much as people currently do in file managers common on most graphical interface systems. One folder will contain, for example, all of the possible targets that will be used in the project. A second folder will contain all the exposures in the project, grouped together into sub-folders for each visit. "Click and drag" techniques will make it easy to move or copy an component, and double-clicking on any component will bring up the interface tool to edit that component. For example, clicking on a target icon should put the user into the Visual Target Tuner with that target already displayed. Later, a user could drag the completed and saved target object onto an exposure calculator and have the target's information automatically incorporated into the calculator.

Figures 12 through 18 contain the proposed window layouts for the first version of the SEA prototype.

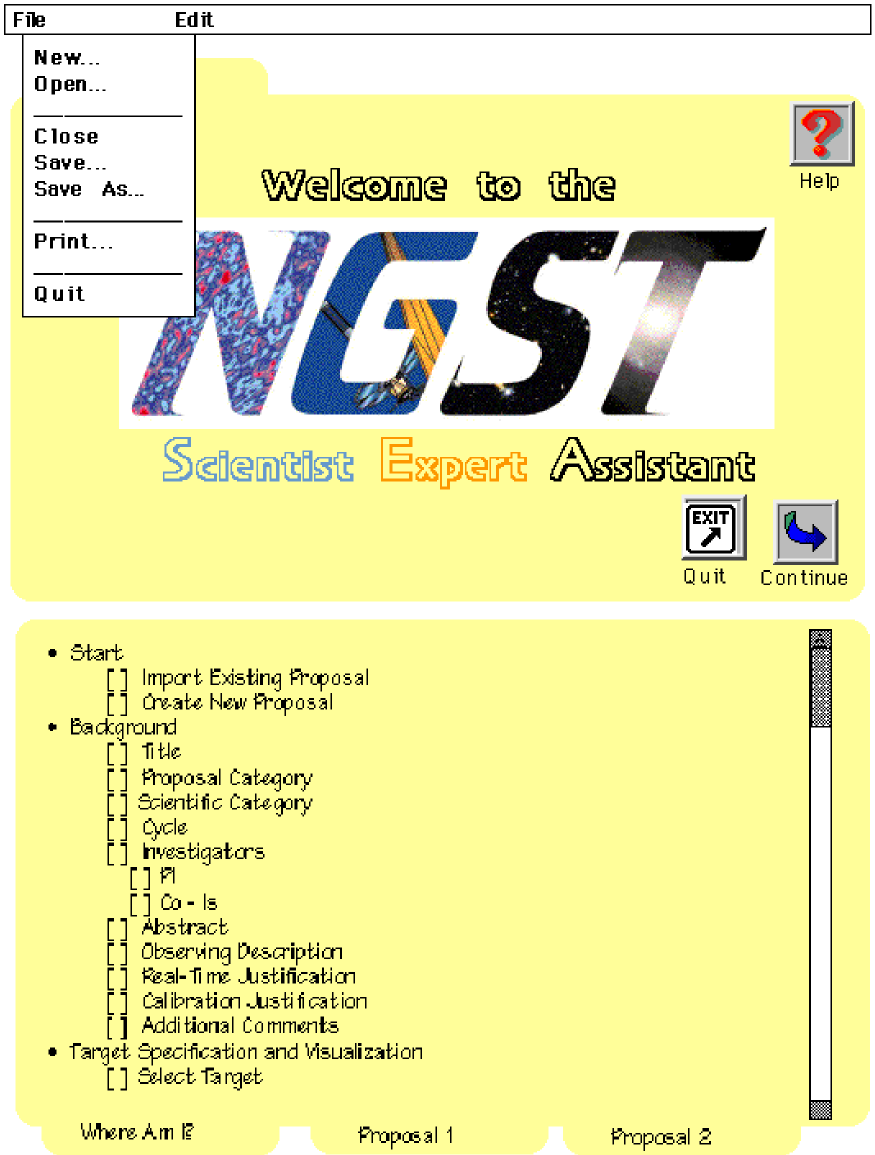

Figure 12: NGST SEA Welcome Window (Top of Options List)

Figure 13: NGST SEA Welcome Window (Bottom of Options List)

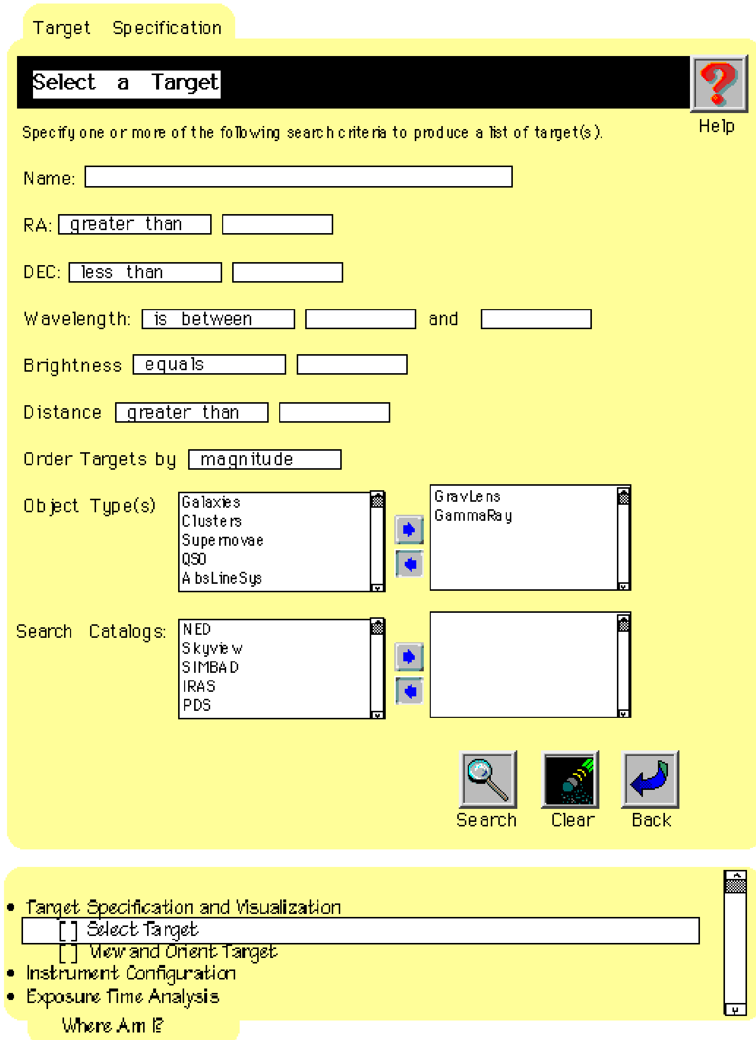

Figure 14: Target Search Parameter Window

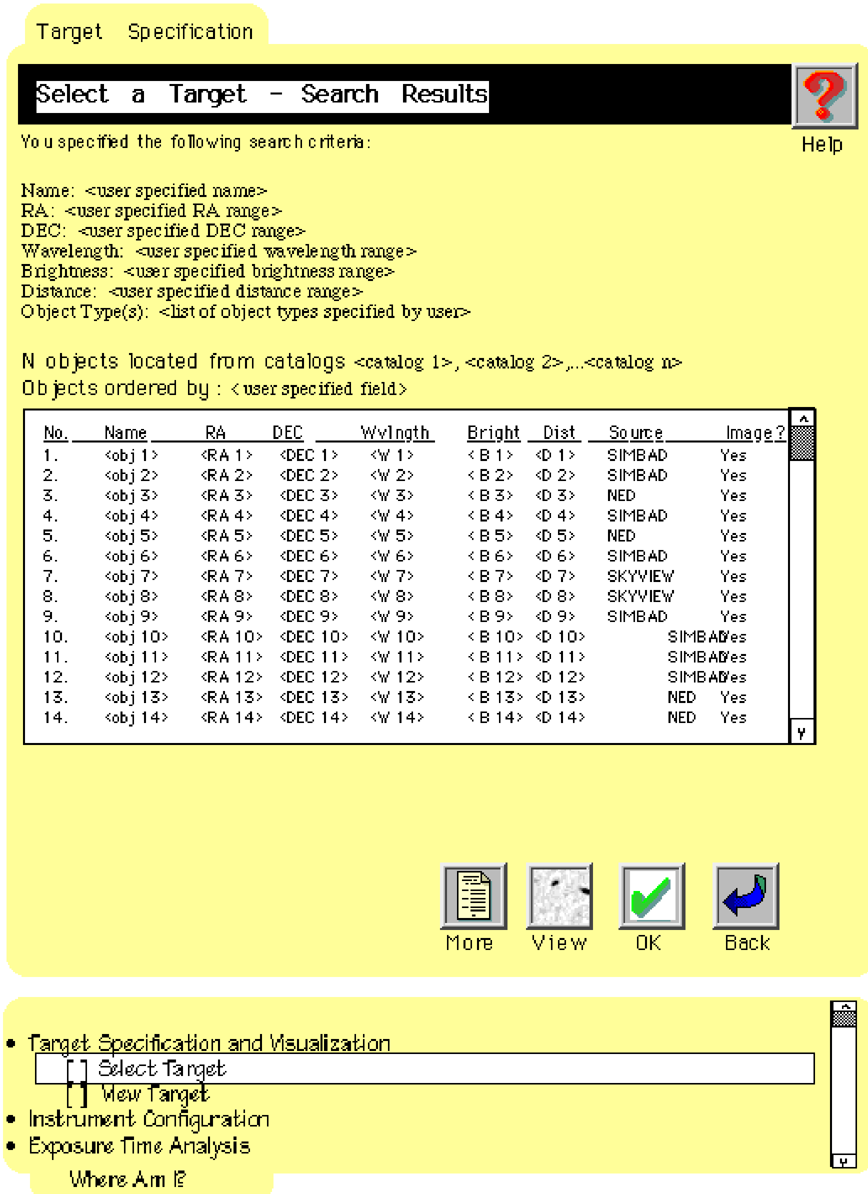

Figure 15: Target Search Results Window

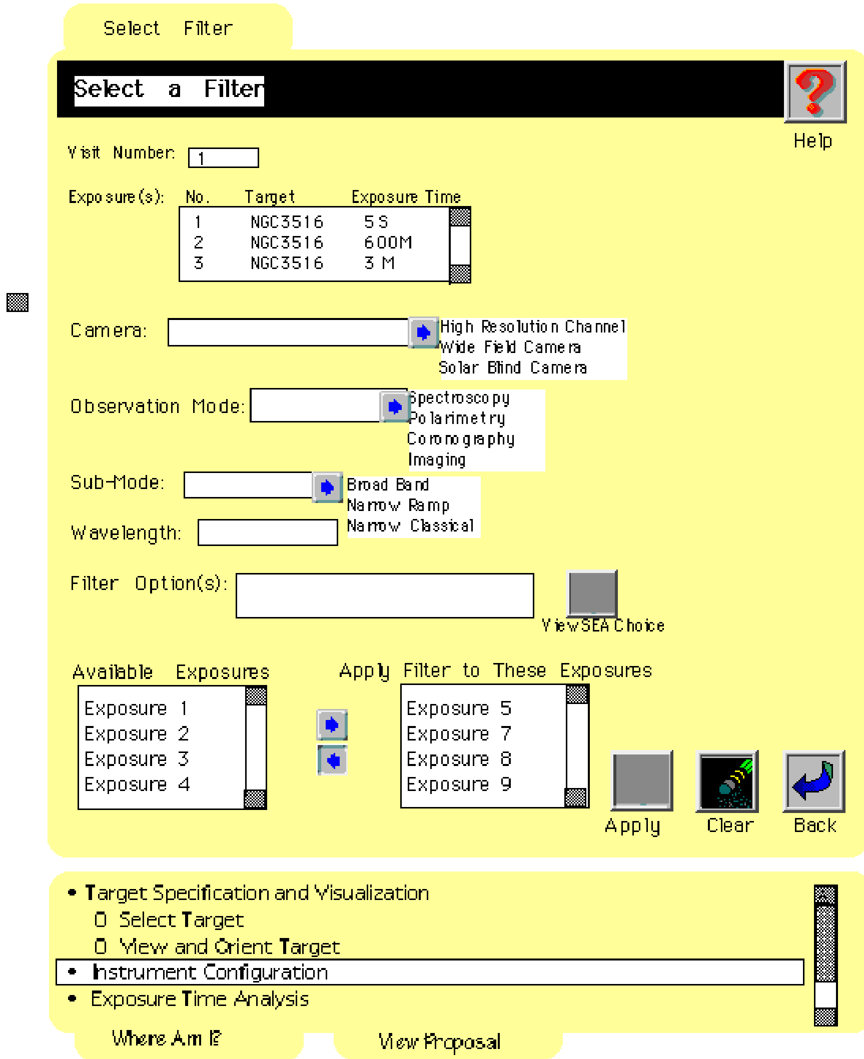

Figure 16: Filter Selection Parameter Entry Window

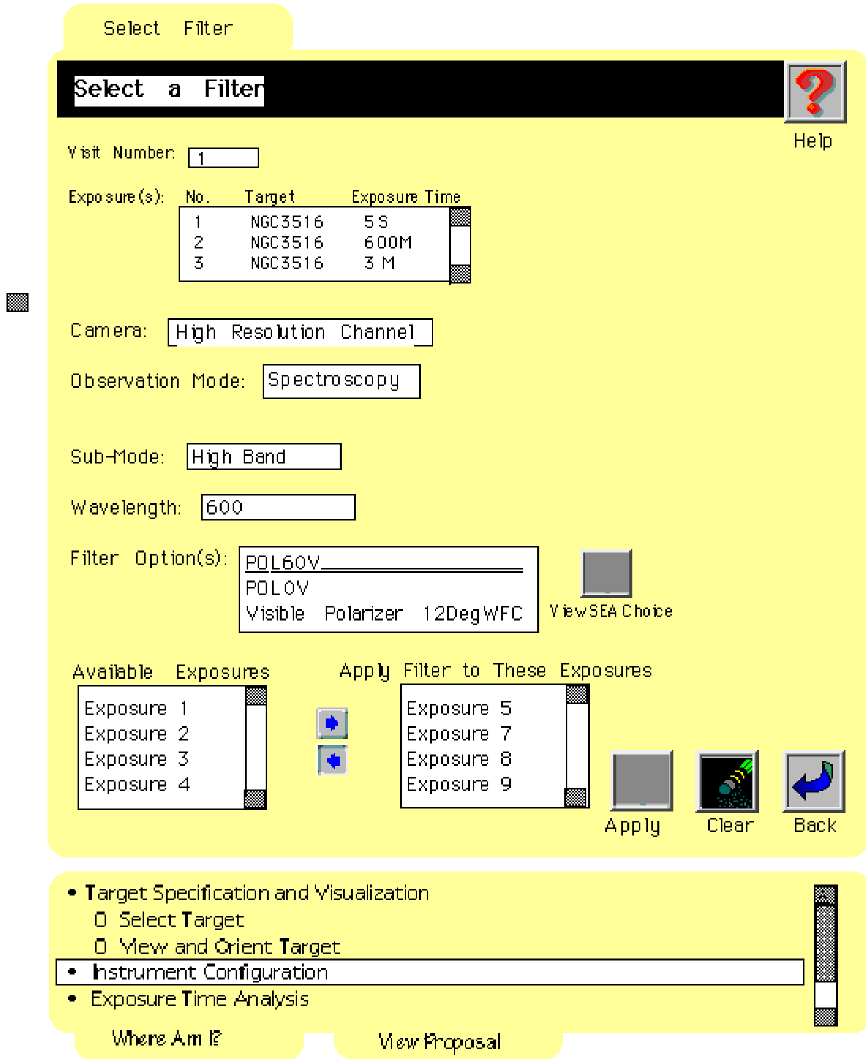

Figure 17: Filter Selection Results Window

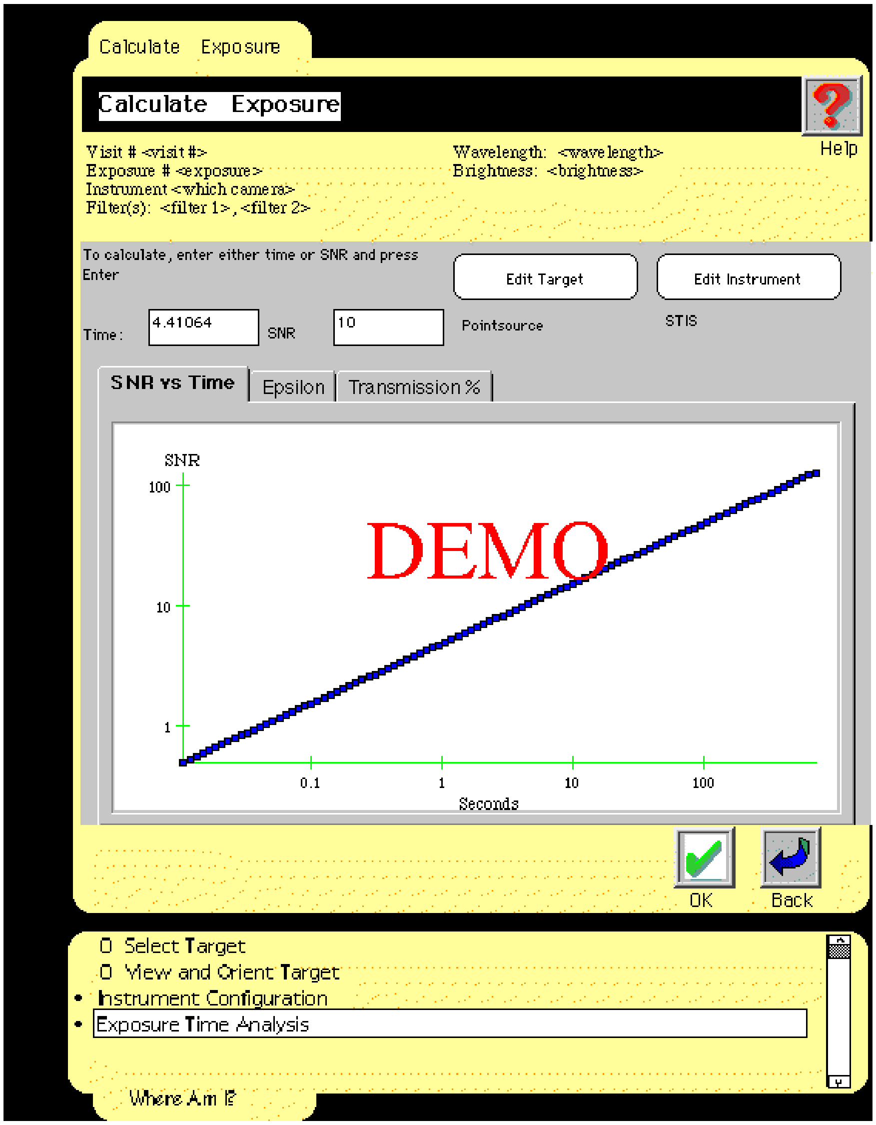

Figure 18: Exposure Calculator Window

4 EXPERT SYSTEM (ES) RECOMMENDATIONS

This section lists the tasks of the HST Phase II proposal process that we feel are most suited to expert system technology. We then present a list of features required by an expert system which may support these tasks, as well as the recommended ES methodology to be used. The various government and commercial tools we examined to support ES development are reviewed, as well as a description of other tools examined. Lastly, the section closes with a description of the Phase II proposal task we have selected to automate with ES technology for the first NGST SEA prototype: namely, the ACS filter selection.

4.1 AREAS OF ES APPLICABILITY

In reviewing the parts of the current Phase II proposal process applicable for expert systems, the following functions apply:

- Determine ACS filter selection;
- Determine which special requirements are needed;
- Based on warnings, provide suggestions on how to make the requested observation valid;
- Give the best and next best options for target selection solutions;
- Check for duplication to see if the image was already taken and included in the archives;
- Interpret some of the images from the visualization tool that is developed;
- In later versions, be able to handle the scheduling of the science experiments.
- In later versions, automatically update proposals as their environment changes over time.

4.2 REQUIREMENTS OF THE ES

Other requirements for the expert system (ES) include:

- The ES should talk first in “science” language and then be converted into “instrument” language as related to ACS targets;
- When calculating exposure times, the goal is to strive for the highest signal/noise by giving the two best scientific solutions;
- The ES should handle two major ACS target concerns—bleeding/spikes and bright sources;
- The ES needs to justify its reasoning;
- The ES should allow for some override features;

- The ES should allow for an object-oriented and rule-based programming approach;
- The ES should work over the web;
- The ES should be easy to use.

In determining the most appropriate AI methodologies for developing this prototype, a rule-based and an object-oriented programming (OOP) approach are most desirable. Both techniques should be used in developing this prototype.

4.3 APPLICABLE ES METHODOLOGIES

A rule-based approach is worth considering when you have experts who are able to specify with a high degree of confidence what they do in specific situations. If a problem is “decomposable”, where the interactions among variables are limited and experts can articulate their decision process with confidence, a rule-based approach is a good candidate and a system may scale well. In the NGST prototype, both these conditions seem to be met.

An object-oriented programming (OOP) approach should also be used as the object paradigm promotes modularity and transparency by providing “self-sufficient” building blocks called “classes.” A class is the definition of how an object will act and store data. The NGST domain lends itself very well to an OOP approach in terms of reducing cost redundancy and promoting a centralization of modular and reusable parts.

The case-based reasoning (CBR) approach is not appropriate for the NGST prototype as it would be very difficult to build a case base for the NGST domain because NGST experiments are different from the Hubble ST experiments. Analogical reasoning would be difficult to use here because the target solutions could not be easily inferred or matched with base solutions (due to the difficulty in populating the case base). Additionally, the NGST instruments are typically re-calibrated which may cause difficulty in building a case base of similar observations. It should be noted, however, that a CBR system can typically be more easily maintained that a rule-based system due to CBR systems being highly scaleable and flexible.

4.4 ES TOOL REVIEW

In reviewing the requirements, functions, and AI methodologies from above, the most appropriate ES shells under consideration for developing the NGST prototype are:

- JESS (Java Expert System Shell)--based on CLIPS (public domain from Sandia National Laboratories);
- Ilog Rules for Java (C++ too)/Ilog Visualization Suite (Ilog, Inc.);
- Elements Advisor/J (Neuron Data)
- ART*Enterprise/Web and ART*Enterprise (Brightware/Inference Corp.).

Each of these expert system development tools will be discussed in the following sub-sections.

4.4.1 JESS

JESS (The Java Expert System Shell) 3.0 is the Java version of CLIPS. JESS is public domain and was developed by Ernest Friedman-Hill at Sandia National Laboratories. JESS is considered to be a work in progress, and as such, a number of features still need to be added. These include: the "and" and "or" conditional elements are not supported on rule LHS (left hand sides); the "test" conditional element is not supported; the only supported slot attribute in JESS is the "default" attribute; and other needed functions. In order to work with JESS, one needs to know CLIPS/Java and this accordingly inhibits the ease of encoding and development as compared with other expert system shells. Additionally, the domain expert would have difficulty in writing his/her own rules using JESS. However, JESS is available at no cost, and the Goddard engineer currently working with the JESS developer has found him very responsive to change requests.

4.4.2 Ilog Rules for Java/C++

Ilog, Inc. has several products that look attractive for the NGST prototype effort. Ilog Rules is a rules engine for C++ and Java. Ilog claims to have the first rule engine in Java. Ilog, Inc. is a well-established company which developed a number of products for the object-oriented, constraint-based reasoning environment (i.e., Ilog Solver/Schedule). Ilog, Inc. also has a Visualization Suite that includes their products, Ilog Views (2D graphics library), Ilog Vision (3D add-on to Views), Ilog InForm (visual database access), and Ilog MultiViewer (extended Model/View/Controller). According to their marketing representative, the following prices for each product are listed below (NOTE: it's not clear if these are government rates, even though we asked):

- Rules (C++ or Java): The developer license for this product is $10K; the runtime* license is $20K.
- Views: The developer license is $10K; the runtime* license is $20K.
- Vision: The developer license is $10K; the runtime license* is $20K.
- Inform: The developer license is $5K; the runtime license price was not quoted.
- MulitViewer: The developer license is $2K; the runtime license price was not quoted.

***refers to per application/project with unlimited users.**

4.4.3 Elements Advisor/J

Elements Advisor/J is by Neuron Data, one of the leading AI companies. Advisor/J is a set of high-performance, advanced class libraries written in 100% Pure Java. It runs on any platform with a Java Virtual Machine and in any browser that supports Java. Advisor/J includes a runtime editor. It directly manipulates any Java or JavaScript object, allowing one to integrate with other applications. The Advisor/J API facilitates the embedding of business rules in an applet, client, or server application. Basically, Advisor/J is a Java rule engine. Elements Expert is the latest version of Nexpert Object, Neuron

Data's rule and object based development tool. The listed price quote given by Neuron for one software development kit of Elements Advisor/J is $6,000.

4.4.4 ART*Enterprise/Web and ART*Enterprise

ART*Enterprise by Brightware Corp./Inference Corp. (one of the oldest established leaders in the AI commercial world) is an expert system development tool that handles object-oriented, rule-based, and case-based reasoning approaches. ART*Enterprise/Web is an add-on component to ART*Enterprise for allowing to connect ART*Enterprise applications to the Web. Queries from Web clients are received via the HTTP protocol, and are parsed by ART*Enterprise/Web into request objects. Request objects are associated with a session ID, to allow easy session management. The ART*Enterprise application analyzes each request object, and generates an appropriate response object. ART*Enterprise/Web takes the completed response object, and merges it with a predefined template to fill in the information for a result page. The result page is tagged with the correct session ID and returned to the browser to continue the session.

ART*Enterprise/Web has the following features:

- Allows ART*Enterprise developers to connect intelligent applications to the Web;
- Provides a high-level interface that frees the developer from having to know HTML, HTTP, Internet session management, network programming, sockets, CGI programming, and C/C++;
- Works with a full range of available Web browsers and Web page authoring tools;
- Plugs into any Web server;
- Runs on Microsoft Windows 95 & NT, and Unix (Solarix, HP-UX, AIX).

The quotes for ART*Enterprise/Web and ART*Enterprise are (according to Jack Lampman, the marketing representative at Brightware Corp.):

Development:

- ART*Enterprise (single user on a single CPU): $10,000
- ART*Enterprise/Web: $ 2,500 Deployment:
- In deployment to single CPUs with one user, the client deployment licensing fee is $1000 per user.
- For deployments that involve the modification or initiation of other processes outside the ART*Enterprise application (Non-Interactive Server or NIS), the fees are driven by the number of processors upon the deployment server (a one process server is $75,000).
- With Client/Server-based deployments, but only providing client interaction to the ART*Enterprise application, these licenses are based upon the number of users (Interactive Server—licensed in blocks of users). For 40 users on the Interactive Server, the licensing fee is $30,000.

Maintenance:

- Annual maintenance for ART*Enterprise development is $3,125
- Annual maintenance for ART*Enterprise/Web is $775.

(Please note that Brightware is not currently on the GSA schedule).

4.4.5 OTHER SOFTWARE TOOLS

In addition to the above recommendations for ES-based work, the team is planning to do web development using the Java language. Java provides a state-of-the-art object oriented language that is developed for platform-independent, web-delivered systems. Currently, the team is using Java 1.1 and Symantec's Visual Café 2.0 as the development environment. Visual Cafe was chosen for its completeness and maturity as an "Integrated Development Environment." While we have some concerns that the product maturity and stability could be improved it remains a significant improvement over the basic Java Development Kit, and is as mature as any of the other products currently on the market.

In addition, where database access and integration is necessary, the team anticipates using Apple's (formerly NeXT) WebObjects software. WebObjects also provides an object-oriented interface, supports Java, and has robust database integration capabilities.

4.4.6 ES Tool Recommendation

At this writing, we are not recommending a specific ES Tool, since we have found further evaluation is necessary. We have seen a demonstration of the Art*Enterprise/Web tool and have submitted a list of questions for the vendor, to which we are still awaiting a response. We plan to schedule a demonstration of the Elements Advisor/J tool, as well as consult with GSFC Code 522 in-house expertise with the JESS environment. We plan to make a final decision in November, 1997. The features of the tools, its interface to the JAVA environment, the user support, and the cost of the tools will be considered in the decision.

4.5 ACS FILTER SELECTION RULE FLOW DIAGRAMS

Part of our Phase I objectives were to identify an initial script that will serve as the basis for initial expert system rules and interface development. We've chosen to use selection of filters for ACS as the initial prototype. This section shows the flowchart the describes the decision process. We recognize that this is, thus far, a very simple rules system. The initial objective is to keep the rule base simple in order to focus on the designing an effective interface. Once we have an interface implemented then we'll expand the rule base to cover the entire process of specifying parameters for ACS.

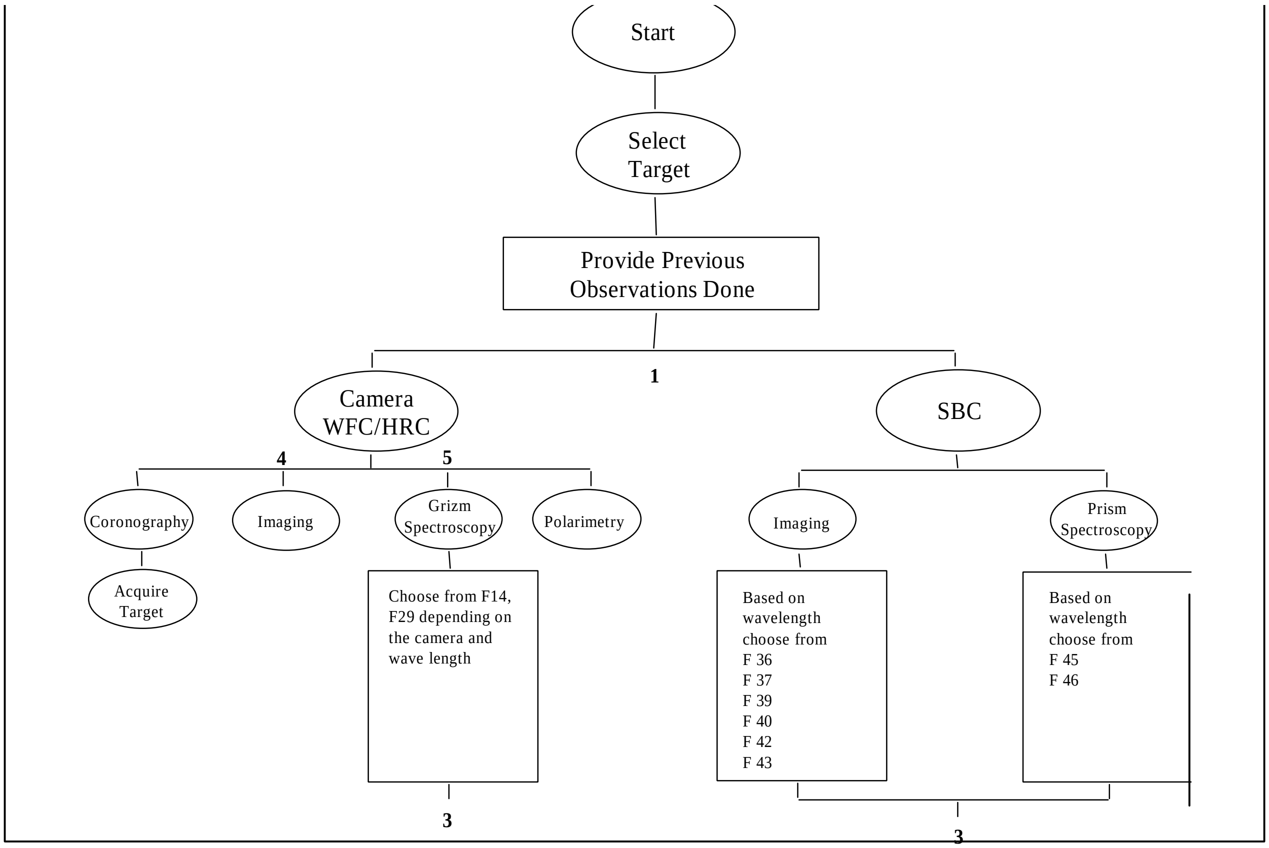

Figure 19: Filter Rule Flow Diagram #1

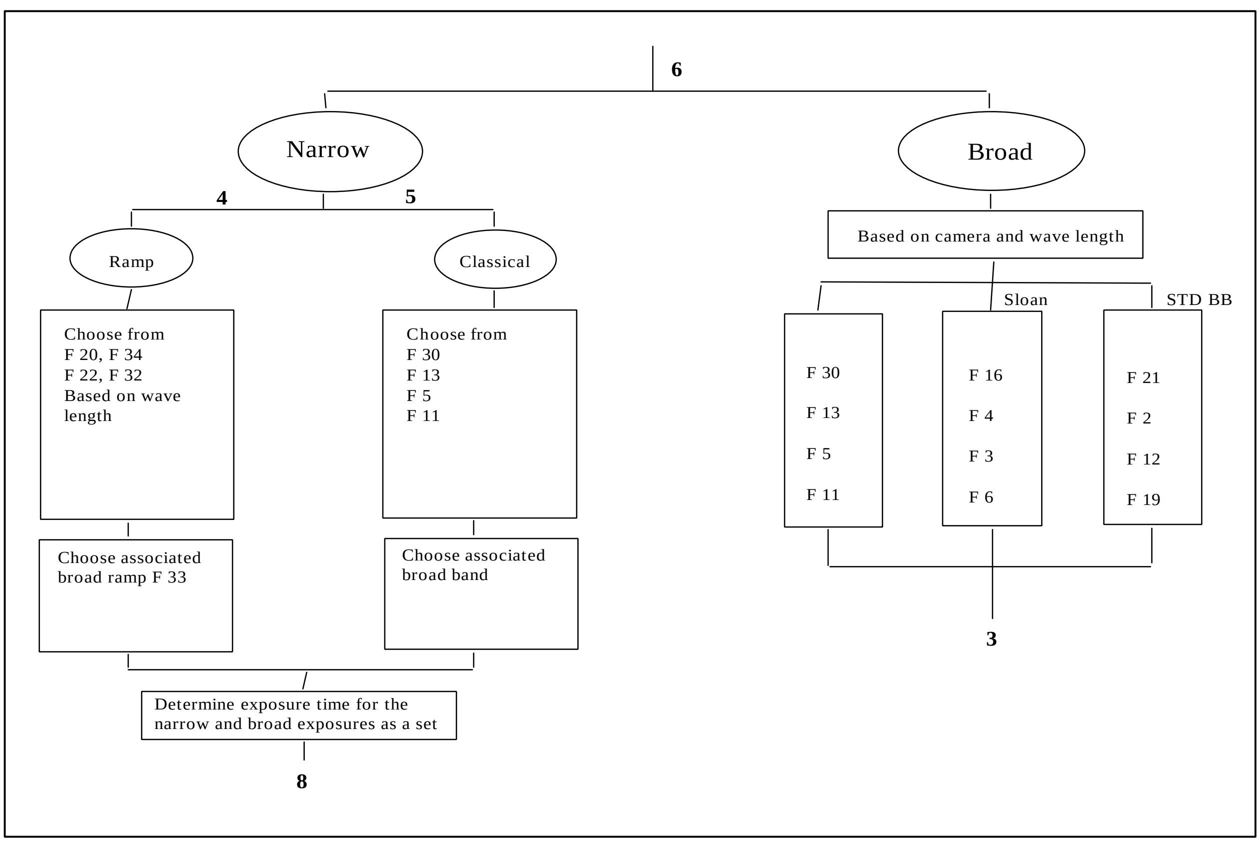

Figure 20: Filter Rule Flow Diagram #2

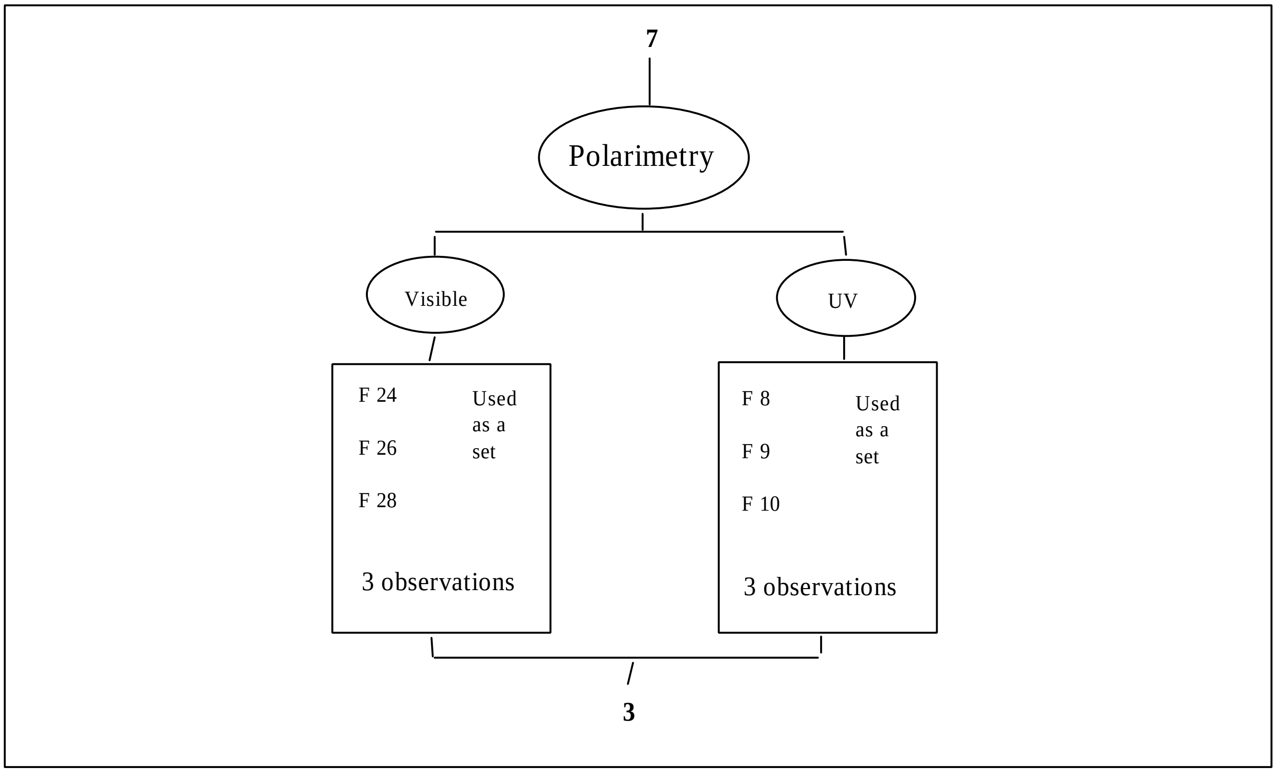

Figure 21: Filter Rule Flow Diagram #3

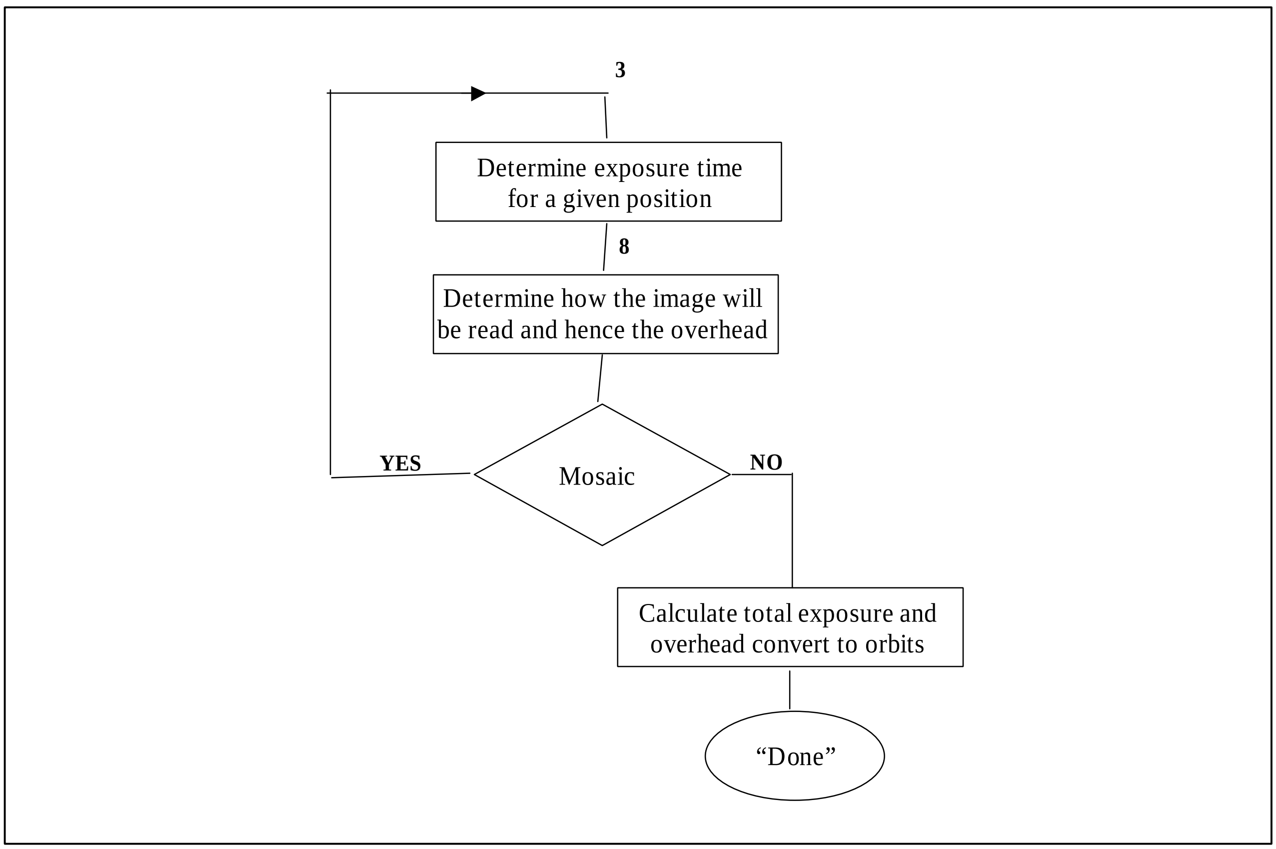

Figure 22: Filter Rule Flow Diagram #4

5 FUTURE GOALS, OBJECTIVES, AND PLANS

The final objective of Phase I was to determine the scope of the project and to develop a project plan for accomplishing our future objectives. During FY 98, we plan to develop the first two builds of several of the modules described above. While FY 97 was mostly spent in research, study, and idea-work, FY 98 begins the development phase. During the next fiscal year, we plan to perform much of the initial design, implementation and testing of the suite of tools discussed above.

A high level development plan appears in section 5.1. Section 5.2 lists the criteria to be used for evaluating the NGST SEA prototype.

5.1 FY 98 PLANS

FY 98 will begin with requirements analysis and design specification, which will form the basis for all SEA development. This work will culminate in a presentation / review. Concurrently, a prototype of the Exposure Calculator will be developed, which will be demonstrated at the SPIE conference in March. Implementation of the rest of the prototype will begin in January, following principles set out in the requirements/design phase. Usability analysis and testing is a critical part of the work to be done. Testing is planned for March, and will repeat iteratively as the prototype evolves.

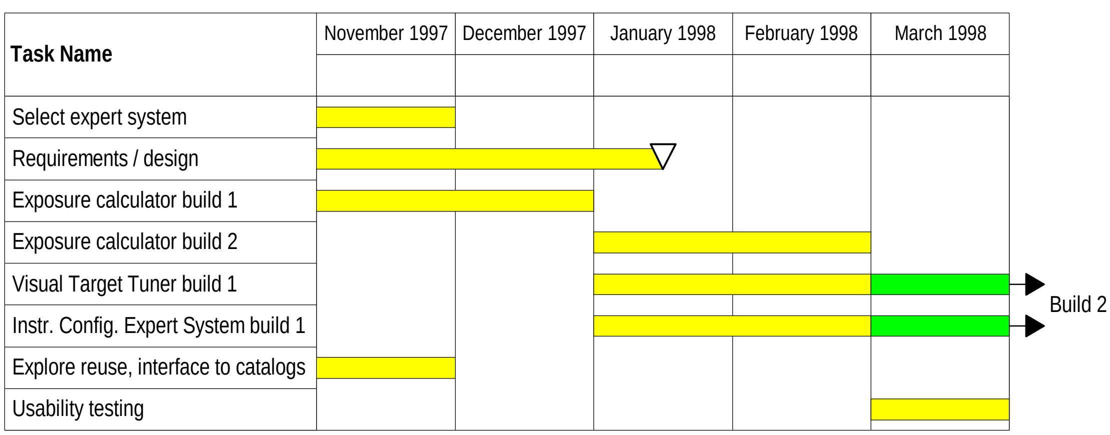

Figure 23: FY 98 Schedule

The following deliverables are planned for FY98:

- Requirements Document
  (Includes Release Plan)
- Design Document
- Test Plan
- Exposure Calculator Prototype Release 1
- NGST SEA Prototype Release 1
- Usability Analysis Reports

5.2 SUCCESS CRITERIA

In order to deem the NGST SEA a success, the following criteria should be realized after the SEA prototype has been in regular use for an evaluation period of 3 months:

- Establish that there has been at least a 50% decrease in the number of submitted proposals requiring corrections by Program Coordinators (PCs) and Contact Scientists (CS)s as a result of syntactical errors.

- Establish that there has been at least a 50% decrease in the number of calls made to Contact Scientists by GOs requesting assistance with target selection.

- Establish that there has been at least a 50% decrease in the number of calls made to CSs by GOs requesting assistance with target orientation.

- GOs report high satisfaction level with overall SEA prototype.

- GOs report a noticeable decrease in the proposal cycle time (from initial creation to final acceptance) as a result of using the SEA prototype.

- GOs report a noticeable decrease in the amount of time consulting instrument handbooks/documentation as a result of using the SEA prototype.

- GOs report a noticeable decrease in the amount of time learning instrument-specific on-line tools (such as exposure calculators.)

- Users report a high level of satisfaction with such SEA usability issues as:

  - - ease of learning tool/accessing help/locating examples

  - - ease in acquiring and installing tool

  - - ease in acquiring and installing updates

- Users report a high level of satisfaction with SEA performance.

5.2 REQUIREMENTS MATRIX

The following matrix shows the order in which features are planned to be implemented.

10/20/97	**NGST SEA Requirement Matrix**				
	(F=Full, P=Partial, E=Enhanced?)				
ID	**Requirement**	**Est Effort**	**Build 1 (3/98)**	**Build 2 (9/98)**	**FY 99**
1	SYSTEM MANAGEMENT AND ENGINEERING				
1.1	Provide an application framework including plug-in OO architecture for new tools				
1.1.1	Basic object definitions		F	E	
1.1.2	Object persistence (able to save and re-read later)			P	F
1.1.3	Object transfer			P	F
1.1.4	Inter-object communication		F	E	
1.2	Science-related class infrastructure		P	F	E
1.3	Provide password protected user access via the Web			F	
1.4	Allow the user to input a proposal summary (ie. Phase I)		P	F	
1.5	Allow the user to edit an existing proposal			P	F
1.6	Completion/Cleanup/Lessons Learned				
2	TARGET SELECTION SYSTEM (TSS)				
2.1	Allow the user to specify the target		F		
2.1.1	by database selection		P	F	
2.1.2	by its name		P	F	
2.1.3	by its coordinates		F		
2.2	Allow the user to specify the instrument to be used		F		
2.3	Completion/Cleanup/Lessons Learned				
3	INSTRUMENT CONFIGURATION EXPERT SYSTEM (ICES)				
3.1	Assist the user in the filter selection process				
3.1.1	Initial small rule set		F		
3.1.2	Expand to full ACS rule set			P	F
3.1.3	Basic automated documentation linkages			P	F
3.2	Completion/Cleanup/Lessons Learned				
4	VISUAL TARGET TUNER (VTT)				
4.1	Display the selected target				
4.1.1	via a canned image from one source		F		
4.1.2	via a canned image from multiple sources		P	F	
4.1.3	via a symbolic image		P	P	F
4.1.4	via a dynamically modeled symbolic image				P
4.2	Allow the user to manipulate the target image				
4.2.1	by sliding and rotating in all directions		P	F	
4.2.2	by zooming in or out			P	F
4.2.3	by identifying objects in a selected region		P	F	
4.3	Provide automated (ES?) target/orientation fine tuning based on				
4.3.1	user selection of areas for inclusion		P	F	E
4.3.2	user selection of areas for exclusion		P	F	E
4.3.3	user selection of areas for placement in the coronalgraph (sp?)				P
4.3.4	minimization of spiking overlap				P
4.3.5	minimization of bleeding overlap				P
4.4	Dynamically display a simulation of the proposed image including				
4.4.1	possible orientations		P	F	
4.4.2	spikes			P	F
4.4.3	bleeding			P	F
4.4.4	noise			P	F
4.4.5	coronalgraph effects			P	F
4.4.6	areas of the sky which are not viewable with the selected ins		P	F	
4.4.7	Other filter effects?				
4.5	Provide intensity graph of user-selected region			P	F
4.6	Multiple exposure/survey layout assistant				
4.6.1	Manual (ability to manually place multiple exposures			P	F
4.6.2	Automated layout based on user outlined area				F
4.7	Completion/Cleanup/Lessons Learned				
5	EXPOSURE CALCULATOR				
5.1	Basic model and graphs		F		
5.2	Expand to include ACS, user manually specifies target info		F		
5.3	Integration with other tools			P	F
5.4	Completion/Cleanup/Lessons Learned			F	
6	PROPOSAL CHANGE/MONITOR AGENT				
6.1	Develop requirements and design			F	

6.1	Develop requirements and design		F		
6.2	Prototype			F	
6.3	Implement ACS Testbed			P	F
6.4	Completion/Cleanup/Lessons Learned				F